mm-22

Indexing Flowers by Color Names using Domain Knowledge-driven Segmentation*

Madirakshi Das R. Manmatha Edward M. Riseman

Department of Computer Science
University of Massachusetts
Amherst, MA 01003
e-mail : mdas,manmatha,riseman@cs.umass.edu

Abstract

We describe a solution to the problem of indexing images of flowers for searching a flower patents database by color. We use a natural language color classification derived from the ISCC-NBS color system and the X Window color names to effectively use the domain knowledge available, provide perceptually correct retrieval and allow natural language queries. We have developed an automatic iterative segmentation algorithm with knowledge-driven feedback to isolate a flower region from the background. The color of the flower is defined by the color names present in the flower region and their relative proportions. The database can be queried by example and by color names. We demonstrate the effectiveness of the strategy on a test database.

1 Introduction

The problem of meaningful retrieval from image databases has generated a great deal of interest in recent years. Most retrieval algorithms have targeted a general image database which may contain diverse types of images [4, 5]. However, there is a growing number of large image databases which are dedicated to specific types and subjects of images. When using general-purpose retrieval strategies on these databases, it is easy to lose sight of characteristics of the domain which could be used to substantially improve the retrieval performance. There may also be special querying requirements in applications in the domain covered by the database.

This work is motivated by the need for indexing a database of flower images which have been digitized from photographs submitted as a part of applications for flower patents to the U.S. Patents and Trademark Office. The aim is to provide a tool to a user who would like to check whether flowers similar in color to a given flower are present in the patents database or be able to list all flowers of a given color by specifying a color name.

Though all images in the database depict flowers, there is no uniformity in the size and location of the flowers in the image or the image backgrounds as shown in Fig 1. There are two main problems to be addressed in this application : the problem of segmenting the flower from the background and the problem of describing the color of the flower in a form which matches human perception and allows flexible querying by example and by natural language color names.
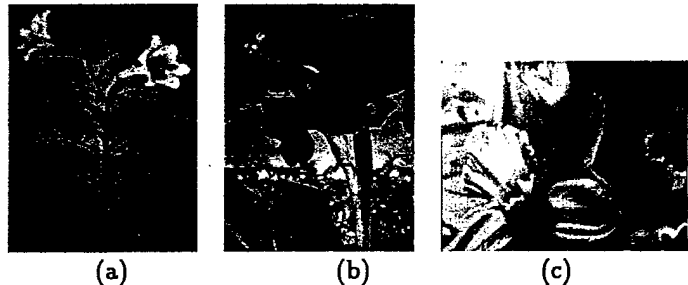

(a) (b) (c)

Figure 1: Example of database images showing different types of backgrounds

We would like to use the characteristics of this domain to automate the segmentation and indexing process. Most of the domain knowledge is in the form of natural language statements; translating these into rules which can be used to build automated algorithms is non-trivial. For example, like most natural subjects, a lot of color-based domain knowledge is known for the flower domain e.g. flowers are rarely green, black, gray or brown in color. Examples of information in other domains would be facts like mammals are rarely blue, violet or green and outdoor scenes often have blue and white skies and green vegetation. However, these types of information can only be used effectively when there is a mapping from the 3D color space to natural language color names. We have constructed a mapping to a natural language color name space using color names from the ISCC-NBS [16] system and the color names defined in the $XWindow^{TM}$ system for this purpose.

We have developed an iterative segmentation algorithm which uses the available domain knowledge to provide a hypothesis marking some color(s) as background color(s) and then testing the hypothesis by eliminating those color(s). The evaluation of the remaining image provides feedback about the correctness of the hypothesis and a new hypothesis is generated when neccessary after restoring the image to its earlier state.

The next section surveys related work and section 3 discusses the mapping from the RGB space to the natural language color names. The domain knowledge available and the segmentation strategy which utilizes it are explained in section 4. Section 5 discusses indexing and retrieval from the database including different types of queries supported. Section 6 describes experiments carried out to test the system and the paper concludes in Section 7.

2 Related Work

There has been work on perceptual organization of the color space in the area of image indexing [1] and in color science [17] without mapping the perceptual groups obtained to natural language color names. These approaches are not very useful in the translation of natural language rules about color into computer usable information. However, they provide good indexing tools when the object of interest has been pre-segmented from the background.

In the reverse approach, color domain knowledge has been mapped to the 3D color space in applications like face identification using skin tones [11] and automatic target recognition, where the part of the color space which corresponds to the object of interest is identified. Modeling the distribution of color points in objects is an important issue in this approach. The set of pixels in each natural object is modeled as a Gaussian PDF in annotating natural scenes in [10]. Regions corresponding to a specified color model are detected in [2].
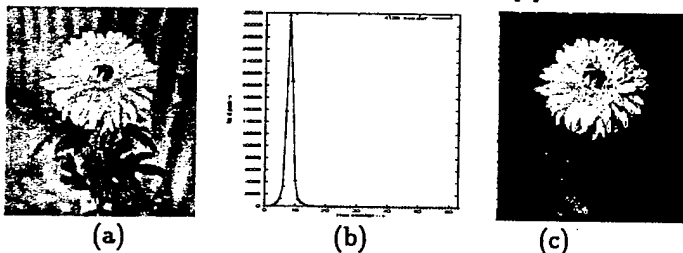

(a)   **(b)**   **(c)**

Figure 2: Use of natural language color classes in background elimination : (a) Original image (b) Hue histogram shows a single cluster (c) Result obtained by deleting black, gray, brown and green from the image

There has been a lot of work in the area of color image segmentation and indexing using histograms [9, 13, 15] in different color spaces [7, 8]. However, these techniques produce color segments which may not correspond to single objects in the scene and also, there is no way of discriminating foreground and background elements. For example, Fig 2 shows a flower image and its hue histogram. Since the hue histogram shows a single cluster, there is no way to use it to isolate the foreground from the background. Eliminating pixels based on their perceived color, on the other hand, leaves only the pixels shown in Fig 2(c), highlighting the usefulness of converting the color space to a color name space based on perception of color.

Automatic foreground/background disambiguation based on multiple features like color, intensity and edge information has been studied in [3], but these techniques work well on relatively smooth backgrounds and objects with sufficient contrast. Image segmentation and querying using blobs based on color and texture is described in [12].

3 Mapping from color space to names

We need tables mapping points on a 3-D color space to color names which should agree with human perception of colors to be useful. We use two sources for names (i) the ISCC-NBS color system which produces a dense map from the Munsell color space to names and the (ii) colors defined by the X Window system which provides a sparse mapping from the RGB space to 359 names. The ISCC-NBS system uses a standard set of base hues and generates 267 color names using hue modifiers. This gives us a color system which can be easily decomposed into a hierarchy of colors where we may use the full color name, partial names, base hues or coarser classes comprised of groups of base hues.

red	green	brown	orange
blue	purple	pink	yellow
violet	black	white	gray

Figure 3: Color classes derived by grouping ISCC-NBS hue names and adding three neutral colors

The color names in ISCC-NBS system often have simpler commonly used alternatives e.g. 'very pale yellowish white' in the ISCC-NBS system is the color 'ivory' and 'light brownish yellow' is the color 'khaki'. The simpler names, like 'ivory' and 'khaki', which often are derived from objects of the same color, are obtained from the definitions in X.

The raw image data available encodes color in the RGB space using 24 bits per pixel. This produces 2^{24} possible colors which is far more than the number of distinct colors that can be perceived by a human. The distances between points in this space are also not representative of the perceived distances between colors. We have used the HSV color space discretized into 64x10x16 bins as an intermediate space to reduce the number of colors as well as have perceptually similar colors in the same neighborhood.

RGB (256x256x256)	(245,195,40)	(233,150,122)
HSV (64x10x16)	(7,8,15)	(2,5,14)
XColor names (359)	goldenrod2	dark salmon
ISCC-NBS colornames (267)	strong yellow	dark brownish pink
Color classes (12)	yellow	pink

Figure 4: Example of color representations used

Each point on the discretized HSV space is mapped to a color defined in X. Points with no exact map are mapped to the nearest color name using the city block measure to compute distances. Each point is also mapped to the ISCC-NBS name [Fig 4]. The ISCC-NBS name is used to produce a color hierarchy so that queries can be general (e.g. blue) or specific (e.g.pale blue). This color structure is also used in segmentation of the flower from its background. Using color names from two sources improves the chances of finding a name which matches the user's natural language query.

4 Iterative segmentation with feedback

We need to segment the regions corresponding to flowers from the rest of the image before we can accurately describe the colors of the flower. The flower regions are isolated from the background using domain knowledge about the color of flowers and also knowledge about the distribution of background regions in photographs.

4.1 Use of domain knowledge

Since we have constructed a mapping from the 3D color space to natural language color names, we can use color-based domain knowledge of the type discussed earlier. We can eliminate most of the frequently occuring elements of the background in flower images by deleting pixels which belong to color classes which do not represent colors of flowers. Black and gray are mostly contributed by the shadow regions in the image, brown pixels come from shadows as well as branches, soil etc. while green pixels are from the foliage and vegetation.

The other observation which is helpful in identifying background regions is that background colors are usually visible along the periphery of the image. If this observation was always true, the background color could be detected with certainty by analysing the colors present in the margins of the image. However, the margins of the image could be of three different types as shown in Fig 1. The flower may be totally embedded in the background, the background and flower regions may interlace along the margins or the flower may fill the whole image.

We can derive some useful guidelines from the fact that the images in the database are photographs depicting flowers. This means that the flower itself will occupy a reasonable part of the image. Also, since the flower is the object of interest, it is unlikely that it will be present *only* near the boundaries of the image. It could, however, be present throughout the image, *including* the boundary region. The background may have other colored objects but they will not usually dominate the main subject, which is the flower.

We also know that the flower images were submitted as part of a patent application. Therefore, we can conclude that there is a single type of flower, though there may be many of them in the image. Due to this, a single prominent segment identified as a flower region can be selected out of multiple segments without loss of information. The goal is to isolate a region in the image from which a good description of the color of the flower can be obtained and *not* the detection of all flower regions in the image.

4.2 Segmentation strategy

Our approach to extracting a region which has a high probability of being a part of a flower is to use the knowledge discussed above in successively eliminating background colors till the remaining region is comprised solely of flower areas. This entails the generation of a hypothesis indentifying the background color(s). However, since the hypothesis may be wrong, we use a feedback mechanism from the segmentation results obtained to redirect our choice of background colors and try a different hypothesis.

We use the *connected components* algorithm whenever we need to identify segments in the image, where each segment is a connected component. The connected components algorithm is run after *binarizing* the image, where the only two classes are pixels which have been eliminated and those that remain.

The outline of the algorithm used to produce a segment from which the flower color is estimated is shown in Fig 6.

The image pixels are labelled by their color classes as well as their nearest X color name. We use a coarse-to-fine strategy when using the color labels - the color class description is used first, finer color name distinctions are used only when neccessary. In the first step, pixels
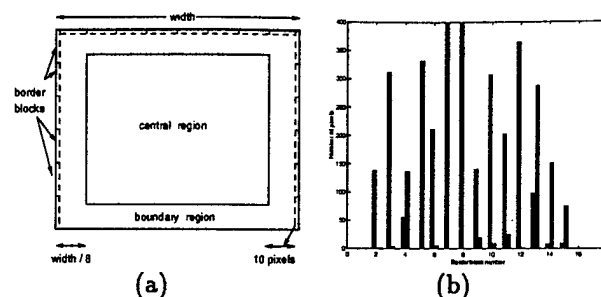

(a)   (b)

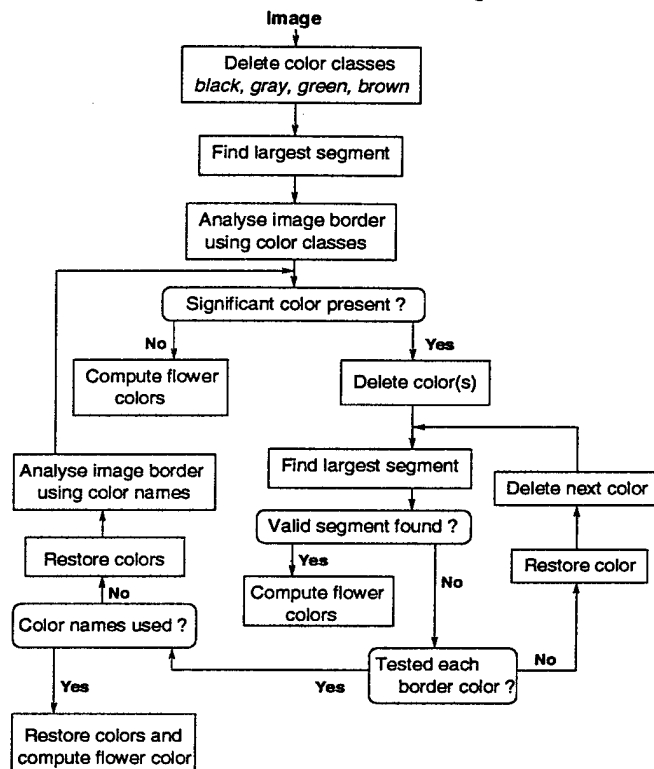Figure 5: (a) Definitions of image regions (b) Color distribution in border blocks of canna image

Figure 6: System Overview

belonging to the color classes *black, gray, brown* and *green* are eliminated since these are non-flower colors and the remaining image is segmented after binarization.

We use two criteria for evaluating whether a segment produced is valid; it should be of a minimum size which is based on the size of the largest segment obtained after deleting the non-flower color classes, and its centroid should fall within the 'central region' of the image as defined in Fig 5 (a). These requirements are based on the domain knowledge discussed in the previous subsection. If there are more than one valid segments, only the largest segment is retained. This step deletes small patches of extraneous colors from other colored objects in the image e.g. the rock in Fig 7. Since we know that the flower is the dominant subject of the image, the largest segment has the highest probability of being a flower region.

Only the pixels comprising the largest valid segment are retained and the rest of the pixels are eliminated. In flower images taken in natural surroundings from a distance, this process is sufficient to produce a good flower
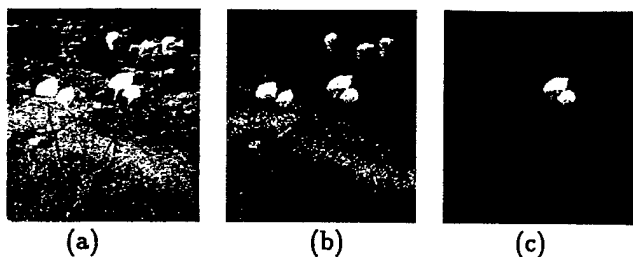
Figure 7: Detecting a reliable flower region : (a) original image (b) image left after deleting non-flower colors (c) largest valid segment

segment. An example is shown in Fig 7 where the final result of segmentation is the image (c).

Further processing is required when the largest segment contains background colors in addition to the flower regions. The presence of background colors is detected by analysing the color composition of the image margins considering the pixels covered by the largest segment only. The margins of the image are divided into border blocks as shown in Fig 5(a). The distribution of color classes in these blocks is computed and colors showing substantial presence in more than half of the blocks are marked as possible background colors. For example, Fig 5(b) shows the color distributions for the two color classes present in the border of the image in Fig 1(b). From this distribution, the color *blue* is marked as a background color since it is present in 11 out of 16 border blocks.
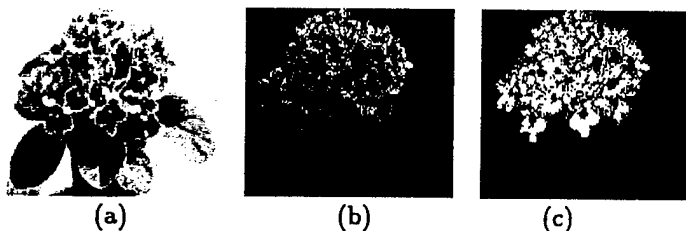

Figure 8: Background elimination : (a) original image (b) image left after deleting non-flower colors and the background color (white) (c) largest valid segment

After eliminating all the pixels belonging to colors which were hypothesized to be background colors, the largest segment in the binarized image is computed. The validity of the segment is tested to determine whether the choice of background colors was correct. Fig 8 shows an example of the final flower segment obtained when the color class *white* was deleted after being correctly identified as a background color.

Our method of detecting background colors is not guaranteed to produce correct results. It will fail for images of the type shown in Fig 1(c), and may also fail for images of the type shown in (b) if there is sufficient overlap between the flower and the margin. An erroneous choice of background color can, in most cases, be detected from the segments generated after eliminating those pixels. In the case of image type (c), the hypothesis for the background color deletes the whole image. In image type (b), if the flower color is deleted instead of the background, only background pixels are left in the image. Since background tends to be scattered among the flower regions and along the margins, no connected components in the central region are usually large enough to be valid, while connected compo-

nents near the boundary do not pass the centroid location test. So, the lack of valid segments is an indicator that the background color selection was wrong.
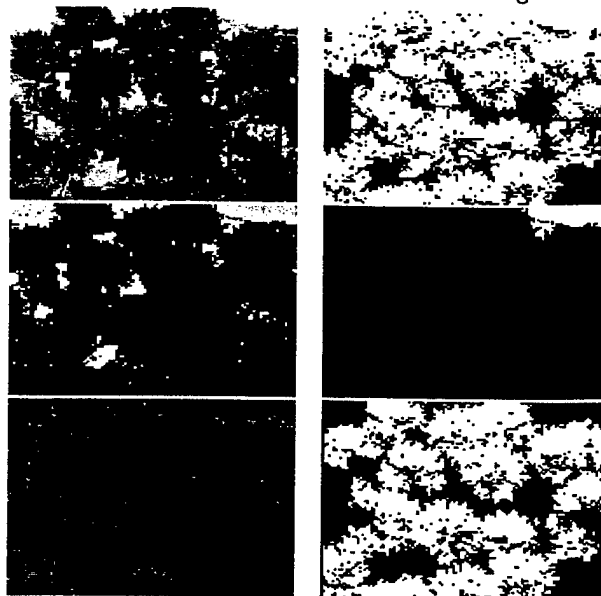

Figure 9: Recovery from erroneous background color selection : (First Row) Original image and segment found after deleting non-flower colors (Second Row) Result of deletion of the color class *purple* which was hypothesized to be a background color and the largest segment obtained (which is not valid) (Third Row) Trying the new hypothesis that the color white is the background color and the valid segment obtained

When feedback is obtained from the segmentation process that the background color chosen was incorrect, the color(s) is restored and the hypothesis that a color is a background color is tested separately, scrolling through each of the colors present in the border region. Fig 9 shows the intermediate steps in detail. From the analysis of the border of the segment obtained first, the color class *purple* is eliminated. This results in a segment whose centroid falls in the boundary region. A valid segment is found when *purple* is restored and another segmentation is carried out after eliminating the new hypothesis for background color, the class *white*.

If no valid segments are found when any of the color classes present in the border are eliminated, one should be able to conclude that the image is of the type in Fig 1(c) and the flowers cover the full image. However, since we are looking at color *classes*, there is an alternative situation where the background is a different shade of the flower color and thus, belongs to the same class. So, we test for this situation by using color *names* to label the pixels instead of the color classes, and repeating the above procedure. An example is shown in Fig 10. When the original image is labeled and segmented, the color class *white* is found to be the background color. However, deleting pixels of the color class *white* deletes the whole image. (The background does not appear to belong to the color class *white* in the figure because the printed colors appear much more saturated than they actually are). When the image is labelled using color names, the colors *HoneyDew* and *MintCream* (which are shades of white) are found from the border block analysis. Deleting these colors leaves the colors *LemonChif-*

fon3 and *Ivory3* which are also shades of white. The remaining image shown in Fig 10(c) produces a valid segment which does not include any background.

When the background cannot be eliminated using any of these trials, the image is assumed to contain only the flower colors and the description is computed from the largest segment obtained after deletion of the non-flower colors.
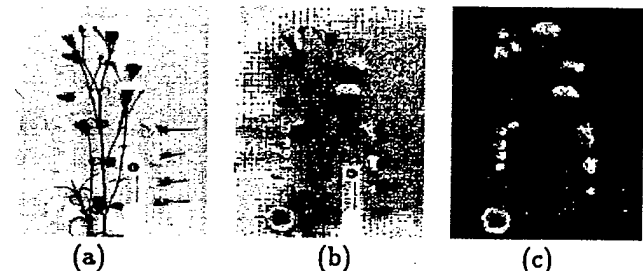

(a)   **(b)**   **(c)**

Figure 10: Using color names for labeling : (a) Original image (b) image left after deleting non-flower colors (c) result of eliminating background colors based on color *names*

The segmentation strategy produces erroneous results only when there are colored objects (excluding the non-flower colors) in the image which are more prominent than the flowers and when the flowers are located only along the margins of the image. Both situations have low probability in the flower patents database.

5 Indexing and Retrieval

The colors present in the segment identified as a flower region in the earlier section are used as features during retrieval from the flower database.

The flower database indexing is based on the types of queries we would like to support. This includes queries using natural language color names. Since there is a wide variety in the names that could be used for querying, the images are indexed by using both the X names and ISCC-NBS color names as keys to improve the likelihood of finding a name supplied by the user as the query in the database index. A third index table is used to access the images by the color classes present in the images.

There are usually more than one color names present in each color class contained in a flower region. The relative proportion of the different shades of the color affects the perceived color in the flower. So the relative proportions of colors in the flower region is also an important factor to be considered.

5.1 Query by name

When a color name is provided as query, the X name index and the NBS color name index are searched for the query color name and its variants. The variants are produced by incompletely specified ISCC-NBS color names and by the X naming system since it uses increasing numbers to indicate darker shades of the original color e.g. 'MediumPurple2', 'MediumPurple3' and 'MediumPurple4' are progressively darker shades of the original color 'MediumPurple'. Since the user is unlikely to know the details of this nomenclature, a query of 'medium purple' should consider all the shades of the color. However, a specific query using one of the defined X or NBS color names could also be issued which will require a knowledge of the valid names. In this case, the exact name is used from the indexes. The retrieved images are ranked by proportion - the flower with a larger proportion of the query color is ranked ahead of a flower with a smaller proportion of the query color. If more than one name is used in the query, a *join* (intersection) of the image lists retrieved for each of the query colors, is returned.

5.2 Query by example

When a flower image is used as a query, the user expects a close color match with the flower shown in the query. In this case, searching for each of the colors present separately and combining the lists often produces poor results. For example, a flower may appear to be a intermediate shade of pink because it consists of a combination of pixels of a darker shade and a lighter shade. Separate retrieval using the two shades present will retrieve a set of flowers which have both these shades, but flowers whose perceived shade does not match the query may be ranked high. This could happen since the relative proportions of the two shades was not taken into account when ranking and therefore, relative proportions of the two shades in the top retrieved flower could be quite different from the query.

In this case, we need to find a distance measure between the query flower and the retrieved flower which takes into account the relative proportions of various shades of a color class in the flower. We do this by computing an 'average' color for each color class present in the query. The HSV coordinates for each X color is computed from its original RGB definition. A weighted average of the HSV coordinates of the X colors present in a color class is computed. The weights are proportional to the relative proportion of the color in the flower segment. e.g. for a flower which has color X1 (h_1, s_1, v_1) and color X2 (h_2, s_2, v_2) in proportion p_1 and p_2 in a class, the average color of the color class is $(p_1h_1 + p_2h_2/p_1 + p_2, p_1s_1 + p_2s_2/p_1 + p_2, p_1v_1 + p_2v_2/p_1 + p_2)$. The retrieved images are now ranked by city-block distance of its average color in each of the color classes from the corresponding query color averages.

6 Experiments

The test flower database we are currently using consists of 250 images. About 100 of the images are from actual flower patents from the U.S. Patent and Trademarks Office. We have added 100 images from CD-ROM collections with complex backgrounds beyond those encountered in images from patent applications to test the segmentation process. The rest are scanned from catalogs of flowering plants and include a few images of colored fruits which are treated the same way as flowers.
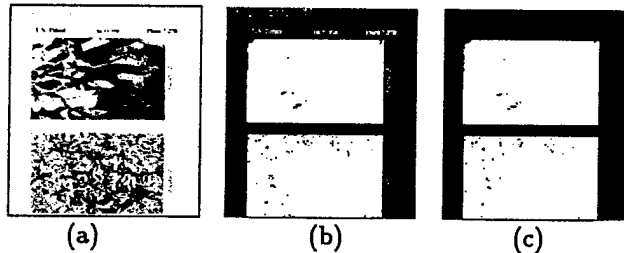

(a)   **(b)**   **(c)**

Figure 11: Detecting images on the patent form : (a) scanned page (b) image left after deleting background color (c) segments found

The pages from the patent forms are of the type shown in Fig 11(a), containing both text and images. Images are detected from the patent forms using the

same strategy of deleting background colors and checking the remaining segments. However, in this case, there may be more than one segments found of significant size as shown in Fig 11(c). These segments are approximated by rectangles and the cropped image corresponding to each segment is added to the database.
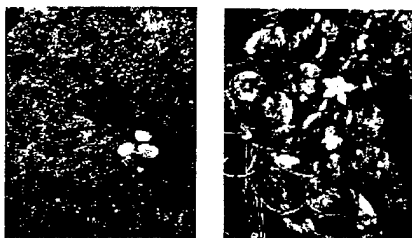

Figure 12: Images on which the segmentation algorithm produces errors

The flower segment identified by the iterative segmentation algorithm was checked for each of the database images and there were only two possibly erroneous segmentation results found for images shown in Fig 12. The segment formed by the pink flowers did not pass the centroid test and the yellow flower region was selected as the most significant segment. This is an image from the CD-ROM collection and unlikely to be a part of a patent application. In the second image, the pale violet leaves of the water lilly constituted the most significant segment which may actually be the correct component of the patent since the flower is given very little emphasis in the image.

We tested the retrieval results obtained using 50 queries of different types and found that the retrieved flowers matched our perception of the natural language color name used in the query or the color of the example flower when an image was used as the query. Fig 13 shows some of the sample retrieval results obtained using different types of queries. The first three rows demonstrate the query by example approach where the first retrieved image was the query image. The fourth row shows the results obtained when using the color 'orange' as query and the last row of images was obtained with the color 'medium purple' as the query.

7 Conclusion

We have proposed a natural language color classification system which can be used to interpret domain knowledge into rules for automatic segmentation of the region of interest from the background. We describe an iterative segmentation algorithm for identifying flower regions. However, our approach can be adapted for any database dedicated to images of known subject about which some domain knowledge is available.

Further work on the current project will include construction of an integrated user interface for the database and tests on a large set of flower images. The use of shape features to distinguish between flowers and color adjacency graphs [14] for distinguishing multi-colored flowers when the number of retrieved images is too large for viewing are also being investigated.

References

[1] T.F. Syeda-Mahmood, "Data and Model-Driven Selection using Color Regions", *Second European Conference on Computer Vision*, pp 115-123, May 1992.

[2] Y. Gong and M. Sakauchi, "Detection of Regions Matching Specified Chromatic Features", *Computer Vision and Image Understanding, Vol 61, No. 2*, pp 263-269, Mar 1995.

[3] Q. Huang, B. Dom, D. Steele, J. Ashley and W. Niblack, "Foreground/background segmentation of color images by integration of multiple cues", *International Conference on Image Processing, Vol 1*, pp 246-249, 1995.

[4] J.R. Bach, C. Fuller, A. Gupta et al, "The Virage Image Search Engine: An open framework for image management", *Proceedings of SPIE, Vol. 2670*, pp 76-87, 1996.

[5] W. Niblack, R. Barber et al, "The QBIC project: Querying Images by Content using Color, Texture and Shape", *Proceedings of SPIE, Vol. 1908*, pp 173-87, Feb 1993.

[6] W. Wang, C. Sun and H. Chao, "Color Image Segmentation and Understanding through Connected Components", *IEEE Conference on Systems, Man and Cybernetics, Vol 2*, pp 1089-1093, Oct 1997.

[7] W. Woelker, "Image Segmentation based on an adaptive 3D-analysis of the CIE-L*a*b* color space", *Proceedings of SPIE, Vol. 2727, Part 3*, pp 1197-1203, 1996.

[8] A.R. Weeks and G.E. Hague, "Color segmentation in the HSI color space using the K-means algorithm", *Proceedings of SPIE, Vol. 3026*, pp 143-154, Feb 1997.

[9] A.F.L. Serafim, "Segmentation of objects in images of natural coloured surfaces", *Proceedings of SPIE, Vol. 2785*, pp 213-222, June 1996.

[10] E. Saber, A.M. Tekalp, R. Eschbach and K. Knox, "Annotation of natural scenes using adaptive color segmentation", *Proceedings of SPIE, Vol. 2421*, pp 72-80, 1995.

[11] Q. Chen, H. Wu and M. Yachida, "Face Detection by Fuzzy Pattern Matching", *Fifth International Conference on Computer Vision*, pp 591-596, June 1995.

[12] S. Belongie, C. Carson, H. Greenspan and J. Malik, "Color- and Texture-Based Segmentation Using EM and Its Application to Content-Based Image Retrieval", *Sixth International Conference on Computer Vision*, pp 675-682, Jan 1998.

[13] M. Das, E.M. Riseman and B.A. Draper, "FOCUS: Searching for Multi-colored Objects in a Diverse Image Database", *IEEE Conference on Computer Vision and Pattern Recognition*, pp 756-761, June 1997.

[14] J. Matas, R. Marik and J. Kittler, "On Representation and Matching of Multi-Coloured Objects", *Fifth International Conference on Computer Vision*, pp 726-732, June 1995.

[15] A. Khotanzad and A. Bouarfa, "Image segmentation by a parallel, non-parametric histogram based clustering algorithm", *Pattern Recognition, Vol. 23, No. 9*, pp 961-973, 1990.

[16] K.L. Kelly and D.B. Judd, "The ISCC-NBS Method of Designating Colors and A Dictionary of Color Names", *National Bureau of Standards Circular 553*, Nov 1, 1955.

[17] G. Wyszecki and W.S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, John Wiley & Sons, Inc. 1982.
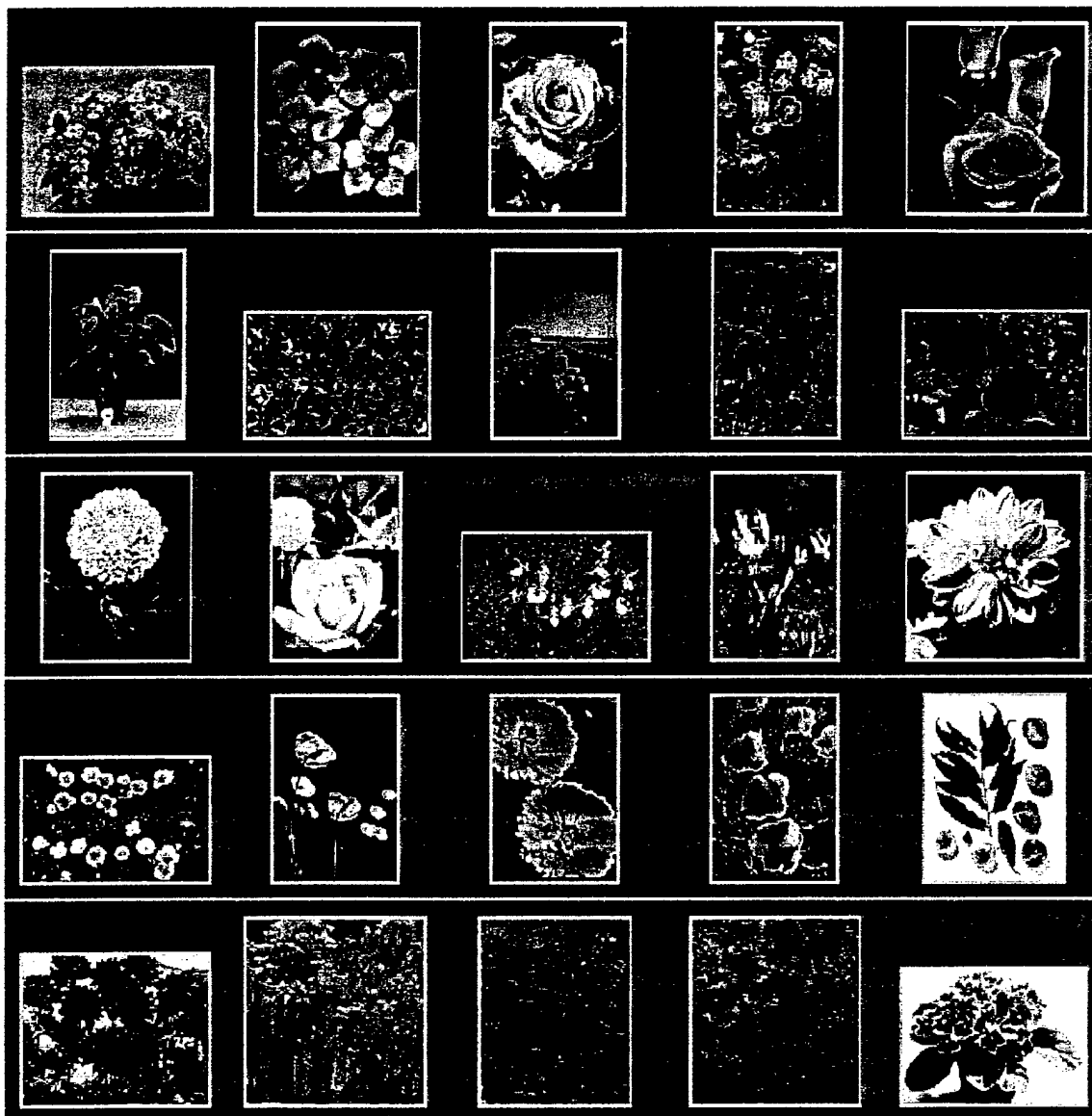
Figure 13: First five retrieved images : Query for rows 1-3 is the first image retrieved in the row, query for row 4 is the color 'orange', query for row 5 is the color name 'medium purple'